

The Software Design for the Wide-Field Infrared Explorer Attitude Control System

Mark O. Anderson
NASA Goddard Space Flight Center
Greenbelt, MD

Kenneth C. Barnes, Charles M. Melhorn, & Tom Phillips
the Hammers Company
Greenbelt, MD

Abstract

The Wide-Field Infrared Explorer (WIRE), currently scheduled for launch in September 1998, is the fifth of five spacecraft in the NASA/Goddard Small Explorer (SMEX) series. This paper presents the design of WIRE's Attitude Control System flight software (ACS FSW). WIRE is a momentum-biased, three-axis stabilized stellar pointer which provides high-accuracy pointing and autonomous acquisition for eight to ten stellar targets per orbit. WIRE's short mission life and limited cryogen supply motivate requirements for Sun and Earth avoidance constraints which are designed to prevent catastrophic instrument damage and to minimize the heat load on the cryostat. The FSW implements autonomous fault detection and handling (FDH) to enforce these instrument constraints and to perform several other checks which insure the safety of the spacecraft.

The ACS FSW implements modules for sensor data processing, attitude determination, attitude control, guide star acquisition, actuator command generation, command/telemetry processing, and FDH. These software components are integrated with a hierarchical control mode managing module that dictates which software components are currently active. The lowest mode in the hierarchy is the "safest" one, in the sense that it utilizes a minimal complement of sensors and actuators to keep the spacecraft in a stable configuration (power and pointing constraints are maintained). As higher modes in the hierarchy are achieved, the various software functions are activated by the mode manager, and an increasing level of attitude control accuracy is provided. If FDH detects a constraint violation or other anomaly, it triggers a safing transition to a lower control mode.

The WIRE ACS FSW satisfies all target acquisition and pointing accuracy requirements, enforces all pointing constraints, provides the ground with a simple means for reconfiguring the system via table load, and meets all the demands of its real-time embedded environment (16 MHz Intel 80386 processor with 80387 coprocessor running under the VRTX operating system). The mode manager organizes and controls all the software modules used to accomplish these goals, and in particular, the FDH module is tightly coupled with the mode manager.

The Software Design for the Wide-Field Infrared Explorer Attitude Control System

Kenneth C. Barnes, Charles M. Melhorn, & Tom Phillips
the Hammers Company

7474 Greenway Center Drive, Suite 710

Greenbelt, Maryland 20770

301-345-5300

kbarnes@hammers.com, cmelhorn@hammers.com, tphillips@hammers.com

Abstract. The Wide-Field Infrared Explorer (WIRE), currently scheduled for launch in September 1998, is the fifth spacecraft in NASA's Small Explorer (SMEX) series. WIRE's mission is to perform a four month survey of galaxies with unusually high star formation rates ("starburst galaxies"). It is a momentum-biased three-axis stabilized stellar pointer which provides high-accuracy pointing and autonomous acquisition for eight to ten targets per orbit. Much of the design is based on previous SMEX missions, WIRE's short mission life and limited cryogen supply impose strict new Sun and Earth avoidance requirements which protect the instrument and preserve cryogen.

This paper presents the design of WIRE's Attitude Control System flight software (ACS FSW), with a concentration on the parts of the design that are new or significantly modified for WIRE. These include its FDH and mode manager modules and its table-driven architecture. The ACS FSW performs all processing necessary for command and control of the spacecraft, and it performs autonomous failure detection and handling (FDH) to insure the safety of the instrument and spacecraft. All these software components are integrated with a control mode manager that dictates which software components are currently active. Lower modes are "safer" because they use a minimal complement of sensors and actuators; as higher control modes are achieved, more software functions are activated by the mode manager, and an increasing level of attitude control accuracy is provided. If a constraint violation is detected by FDH, a safing transition to a lower control mode is triggered. The WIRE ACS FSW satisfies all of its target acquisition, constraint checking, and pointing requirements, and it provides the ground with a simple means for reconfiguring system parameters via table load.

I – Introduction

Laboratory. The spacecraft is being built at NASA's Goddard Space Flight Center.

WIRE Mission Overview

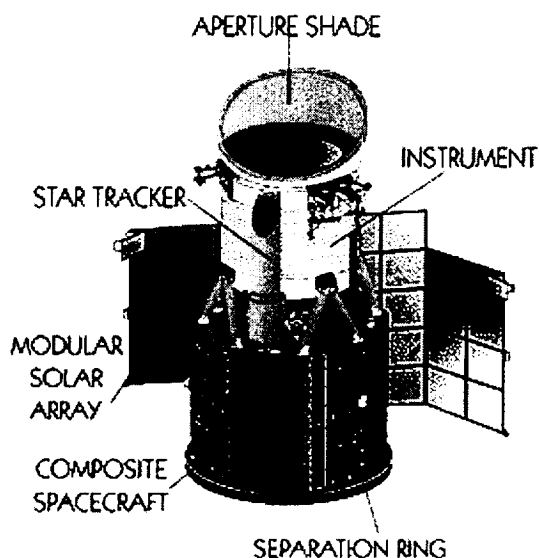
The Wide-Field Infrared Explorer (see Figure 1) is the fifth of five spacecraft in the NASA Small Explorer (SMEX) series. Its predecessors were the Solar Anomalous Magnetospheric Particle Explorer (SAMPEX – launched in 1992), the Fast Auroral Snapshot Explorer (FAST – launched in 1996), the Submillimeter Wave Astronomy Satellite (SWAS – currently waiting to launch), and the Transition Region and Coronal Explorer (TRACE – launched in 1998). WIRE's primary objective is to perform a survey of "starburst" galaxies, which emit most of their energy in the far infrared end of the spectrum. The instrument is being developed by a joint team from the Jet Propulsion Laboratory and Utah State University/Space Dynamics

WIRE Survey

The WIRE survey will cover over 100 square degrees of sky, and will be conducted using a cryogenically-cooled 30 cm imaging telescope. Astronomical sources will be detected in the 12 and 25 μm bands at very faint flux levels. WIRE will be placed in a circular 540 km orbit with an inclination of 97.55 degrees such that observations can be made at high galactic latitudes away from the ecliptic plane (since the goal is to observe faint sources outside our Milky Way Galaxy). Up to four primary science targets will be observed each orbit, and each observation segment will last approximately ten minutes. Each observation segment will be broken into several short exposures (on the order of one minute each) of the same target field; these short staring exposures will be separated by a small

positional offset, or “dither” (on the order of one arcminute). To avoid pointing the telescope too near the Sun or Earth (or into the velocity vector early in the mission), each orbit will contain several secondary observation segments, for a total of approximately 8 to 10 (possibly 12) targets per orbit (the upper bound on the number of targets processed per orbit is limited by the daily uplink volume allowed). The WIRE survey is expected to find 10,000 to 30,000 starburst galaxies or more, and the data will be used to study their evolutionary history out to redshifts of 0.5 - 1.0. It is also expected to reveal the evolutionary history of extremely luminous galaxies beyond redshifts of 5.

Figure 1: WIRE Spacecraft



ACS Pointing Requirements

To accomplish WIRE’s scientific objectives, the attitude control system (ACS) must satisfy several functional and performance requirements. First, to support science data collection, the ACS must provide an attitude control mode which slews and points the spacecraft in accordance with a series of uplinked targeting parameters (a science “timeline”). This mode is called Stellar Point (STP), and is implemented in the ACS flight software (FSW). It allows for the acquisition and processing of the two types of targets required by the science objectives – “fixed” and “dither”. Processing for fixed targets involves continuous staring until it is time to slew to the next one

in the timeline. Processing for dither targets involves periodically performing small offset slews around a target in a grid fashion. Once in STP mode, the spacecraft should remain there for the remainder of the mission. Each new target in the timeline causes the ACS to terminate processing of the current one and perform a slew to acquire the new one. This process of slewing from target to target will be repeated indefinitely as long as valid targets are uplinked in the timeline and no pointing constraints are violated. STP mode must satisfy the performance requirements specified in Table 1.

Table 1: Stellar Point Performance Requirements

Pointing Accuracy	3 arcmin (3σ)
Azimuth Accuracy	28.5 arcmin (3σ)
Pointing Stability	6 arcsec per 64-sec exposure and 12 arcsec for more than 86% of the time
Slew Time	72 deg in 3 min
Dither Time	60 arcsec (3σ) in 7 sec

In order to insure the health of the instrument and to preserve cryogen, strict pointing constraints must be enforced by the ACS in all control modes so that the boresight of the instrument doesn’t point too closely to the Sun or Earth (see Table 2).

Table 2: Pointing Constraint Requirements

Sun Avoidance Constraint	Instrument boresight more than 75 degrees from Sun line
Earth Avoidance Constraint	Instrument boresight within 30 degrees of zenith

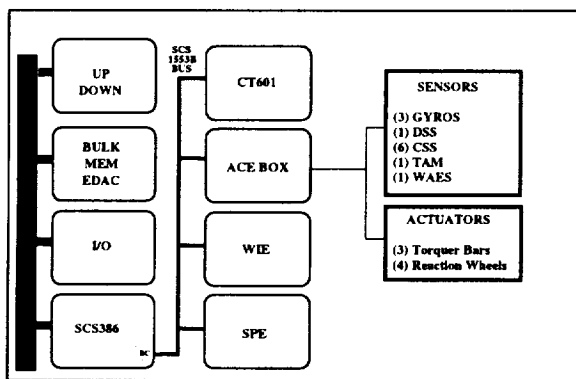
II – Attitude Control Subsystem Overview

ACS Components

To provide three-axis stabilized control, the ACS requires appropriate sensors, actuators, computers, and

the associated infrastructure necessary for internal and external communication. The relationship between these components is depicted in the context diagram in Figure 2. The WIRE ACS hardware complement includes the following sensors and actuators: four reaction/momentum wheels, three mutually perpendicular magnetic torquer bars, three two-axis tuned restraint inertial gyros, one three-axis flux gate magnetometer (TAM), six coarse sun sensors (CSS), one digital sun sensor (DSS) and one wide-angle Earth sensor (WAES). All of these components are driven by the Attitude Control Electronics (ACE), which contains the electronics and software necessary to process sensor information and to command the actuators. All communication with these components is done through the ACE box. The ACS has one more sensor, a CT-601 star tracker, that is not driven by the ACE.

Figure 2: WIRE ACS Context Diagram



The ACS FSW resides in the Spacecraft Computer System (SCS), which consists of a 386 microprocessor running at 16 MHz, a 80387 math coprocessor, 448K EEPROM, 1MB RAM, and a UTM 1553 communications chip. The SCS386 contains the Command and Data Handling (C&DH) software and the ACS software, and these execute under the VRTX real time operating system.

Communication between major ACS components is accomplished via a MIL-STD-1553 bus. As shown in Figure 2, the SCS processor is configured as the bus controller, and the ACE, the CT-601 star tracker, the Spacecraft Power Electronics (SPE), and the WIRE Instrument Electronics (WIE) are all configured as remote terminals.

ACS Modes

In addition to the requirements for processing science targets in STP mode, the ACS must meet many derived requirements which call for several lower control modes¹. These modes satisfy power and thermal requirements, maintain the required system momentum bias, provide intermediate modes for transitioning into STP, and provide safe fall-back modes for handling constraint violations, hardware failures, and other anomalies. These modes are implemented in various ways within the ACS; one hardware-only mode is built into the Attitude Control Electronics box (ACE), one software mode is implemented in the ACE's computer, and five software modes are implemented in the Spacecraft Computer System (SCS) processor. Though all of these modes are part of the ACS, the ACE and SCS processors are distinct, and their respective control modes are independent. The software running in the two computers is distinguished by referring to the "ACE FSW" and the "ACS FSW".

The ACE contains an analog controller which commands the spacecraft attitude during initial operations. This mode is a hardware-only mode called Analog Acquisition, and it is the default state of the ACE upon power-up. This mode is not used subsequent to initial operations unless a severe anomaly occurs, because it does not perform the Earth avoidance control necessary to protect the instrument once its cover has been ejected. This mode damps spacecraft body rates and keeps the spacecraft power positive. The ACE's software mode is called ACE Safehold (ACESH), and it is considered the lowest safe mode. This mode takes control of the spacecraft if there is a failure in the SCS Processor or a problem with its 1553 communications link. It uses the smallest possible complement of sensors and actuators to keep the spacecraft power positive and the instrument thermally safe. To accomplish this, it does not depend on the health of the SCS processor or the 1553 bus.

The ACS FSW provides five control modes. This paper concentrates on the design of these modes, which are outlined in Table 3. They are SCS Safehold (SCSSH), Magnetic Calibration (MCAL), Zenith Sunpoint (ZSP), Transitional Stellar Acquisition (TSA), and Stellar Point (STP). Whenever the ACE is not controlling the spacecraft, one of these modes must be active. SCSSH performs the same control law implemented in ACESH;

Table 3: ACS Software Modes

Mode	Sensors	Actuators	Attitude Determination	Attitude Control	Target Method
SCS Safehold (SCSSH)	CSS, DSS, TAM, WAES	MTB, Y-RW or ABC-RW	None	B-dot, Y-Sun precession, Earth avoidance, momentum bias (same as ACESH)	None
Magnetic Calibration (MCAL)	TAM	MTB	None	None	None
Zenith Sunpoint (ZSP)	CSS, DSS, TAM, Gyros	MTB, 4 RW's	Coarse	3-axis stabilized, momentum management	Continuous zenith pointing
Transitional Stellar Acquisition (TSA)	CSS, DSS, TAM, Gyros, CT-601	MTB, 4 RW's	Coarse	3-axis stabilized, momentum management	Initial timeline target acquisition using CT-601
Stellar Point (STP)	CT-601, Gyros	MTB, 4 RW's	Fine	3-axis stabilized, momentum management	Fixed or Dither timeline pointing

it serves as an independent backup and a graceful way to switch control between the ACE processor and the SCS processor. MCAL mode is used to calibrate the ACS software by removing the torque rods' magnetic contamination sensed by the three axis magnetometer. This mode does not actually perform any attitude control while the short calibration sequence executes.

ZSP, TSA, and STP modes all execute the same three-axis stabilized control law, and they all maintain the desired system momentum bias while unloading unwanted momentum from the reaction wheels. They differ in their means of determining attitude and generating target quaternions. ZSP and TSA modes use only Sun sensor, magnetometer, and gyro data to compute the spacecraft's "coarse" attitude, while STP mode has the additional input from the star tracker to generate a "fine" attitude solution. In ZSP mode, the models data are used to continually compute a zenith target quaternion, while TSA and STP modes use information from the science timeline to compute the desired target quaternion. TSA mode is used to transition gracefully from ZSP to STP and enter the

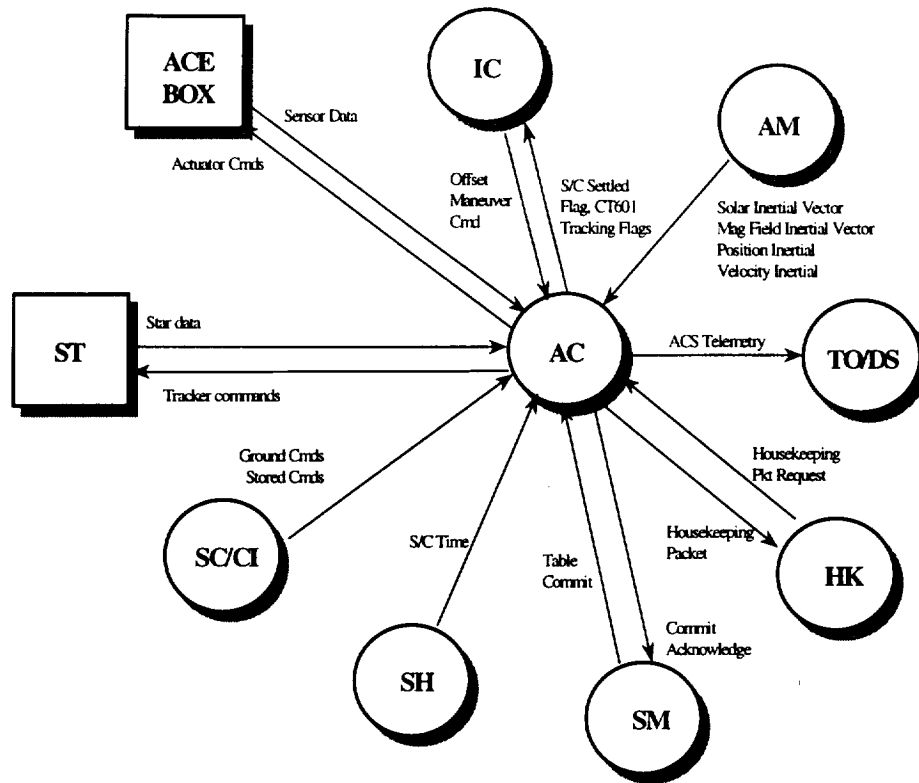
timeline. This is a special case because the software attempts to acquire the "initial acquisition" science target using only the attitude determined by the coarse estimator. Subsequent targets are then acquired in STP mode with the benefit of the fine attitude.

III – Attitude Control Software Design

ACS Software Components

The ACS FSW consists of two distinct tasks and a library of math routines; these are the Attitude Control (AC) task, the Attitude Models (AM) task, and the Attitude Control Math Library (AL). These constitute just three pieces of the software operating in the SCS processor; the remainder of the tasks belong to the C&DH software subsystem. The ACS FSW has important interfaces with several of these C&DH tasks, as shown in Figure 3. In addition, the ACS FSW communicates with two electronics boxes via the 1553 bus. These are the ACE and the CT-601, which provide raw sensor data to drive the ACS FSW control loop.

Figure 3: ACS Software Task Interfaces



ACS FSW Interfaces with C&DH Tasks

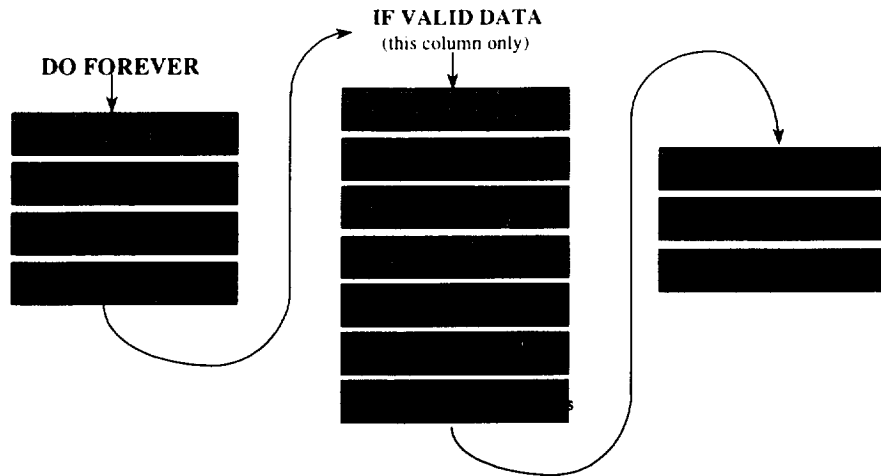
In order to support the dither science observation mode, the AC task must communicate with the Instrument Controller (IC) task in the C&DH. AC sends IC a telemetry packet (the ACS State Packet) which contains all of the information required to manage the current observation. This packet includes a variety of ACS status flags, among them the ACS “settled at target” flag, the CT-601 star tracking indicators, and the current ACS control mode. Other C&DH tasks closely coupled with the AC task include the Stored Command Processor (SC) and Command Ingest (CI) tasks for routing ground commands to AC, the Scheduler (SH) task for providing the system time to AC, the Software Manager (SM) task for handling table loads, the Housekeeping (HK) task for monitoring AC’s health, and the Telemetry Output (TO) and Data Storage

(DS) tasks for routing AC telemetry to the ground and bulk memory. Not shown in Figure 3 is the interface with the Software Bus (SB) task, which provides utility routines for sending actuator commands across the 1553 bus to the ACE box.

ACS Software Architecture

Execution of the AC task is driven by receipt of raw sensor data from the ACE. This data is scheduled to arrive every 100 ms, and each arriving packet causes AC to execute one cycle; therefore, the AC control cycle is 10 Hz. AC implements modules for sensor data processing, attitude determination, attitude control, guide star acquisition, actuator command generation, command processing, telemetry processing, FDH, and mode management. These modules execute each control cycle in the manner specified in Figure 4.

Figure 4: ACS Software Control Loop



ACS Software Heritage

The software development philosophy for the SMEX program has stressed modular design and code reuse. Therefore, despite differences in each spacecraft's requirements and science objectives, a significant part of each software implementation has been brought forward from previous missions in the program. Rather than just adding, modifying, or removing modules to support a particular mission, each implementation has incorporated general improvements into the overall design. The WIRE ACS FSW inherited much of its architecture from TRACE, since that was the most recent prior mission, but the SWAS science requirements more closely resemble those of WIRE, so much of the SWAS FSW was reused as well. Several modules were added or modified to accommodate WIRE-specific requirements, and several changes were made with an eye towards improving code modularity, easing software maintenance, simplifying operational procedures, and making the software more configurable. Table 4 summarizes WIRE's code heritage and new features. Other elements reused from SWAS/TRACE with virtually no change are the AM task, the attitude determination module, the attitude control module, and the telemetry generation module. Most of the inherited parts of the ACS FSW have been detailed elsewhere², so the remainder of this section will concentrate on WIRE's new features.

Table 4: ACS FSW Heritage and Evolution

SWAS/TRACE Features	New WIRE Features
SWAS control mode scheme	Mode manager to handle ground-commanded, FDH-initiated, and autonomous mode transitions
SWAS/TRACE sensor and actuator data processing	WAES data processing
SWAS/TRACE Digital Sunpoint Mode	SCSSH mode with WAES-based Earth avoidance
Basic FDH logic scheme, including Sun constraint	WIRE Sun and Earth avoidance constraints
SWAS star acquisition algorithm	Star acq enhancements to handle dropouts (LOTs)
Basic SWAS instrument interface	Dither target processing
Software configuration by ground command	Table-driven design with much more configurability, many fewer commands
Total: ~85% of code	Total: ~15% of code

IV - ACS Mode Manager Design

Mode Manager Overview

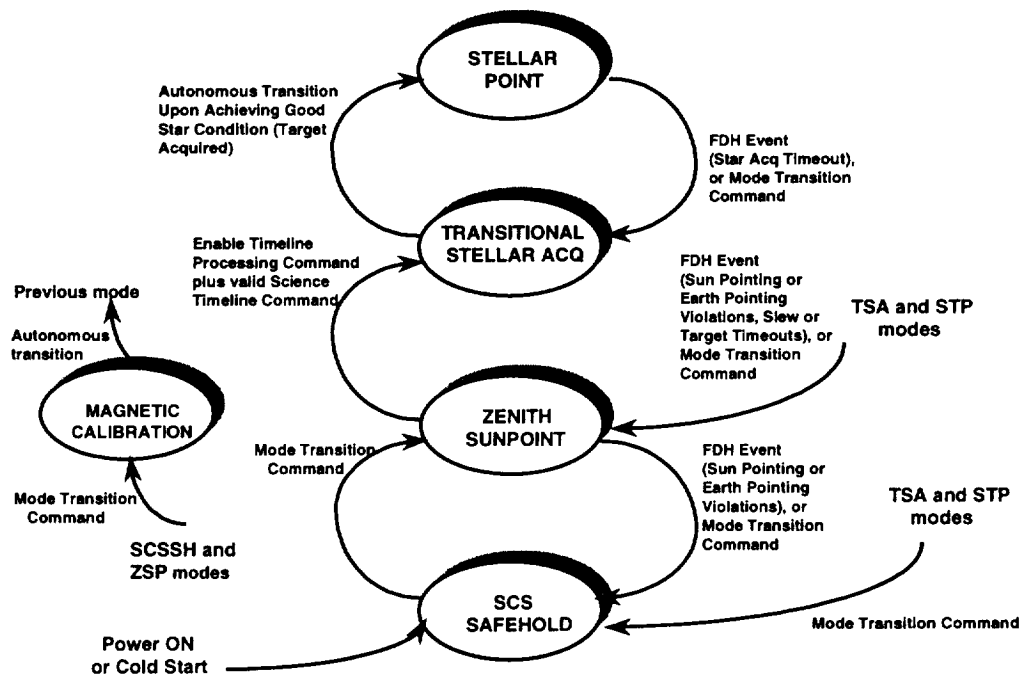
The WIRE ACS FSW must always be in one of five states – SCSSH, MCAL, ZSP, TSA, or STP. These control modes are ordered from lowest to highest in a hierarchy of control pointing accuracy. Lower modes generally have fewer sensors available and use a minimal complement of actuators to control the spacecraft attitude. Transitions between modes may be commanded by the ground, they may be caused by an FDH violation, and they may be initiated autonomously by mode manager itself. More than one of these transition types may occur on a given cycle. The function of the mode manager is to process all such mode change requests, resolve conflicts, and take appropriate action. All defined mode transitions are depicted in Figure 5.

Most of the other ACS FSW modules are critically linked with the mode manager module; by virtue of a mode change, whole sections of the code are activated or deactivated. Some modules (FDH, for example) are only partially executed in some modes, or they are run with a different set of parameters.

Mode Manager Functional Design

The mode manager design was chosen to fit into the pre-existing SWAS/TRACE software architecture. It incorporates many of the same control flags and mechanisms, but it groups all the mode setup logic into one module for ease of maintenance. It executes at the end of each control cycle to configure the software for the next one. It handles multiple transition requests on a given cycle by using a predetermined prioritization scheme. Autonomous mode changes initiated within mode manager have the lowest priority, mode changes commanded from the ground have medium priority, and FDH-initiated mode changes have the highest priority. This is because all FDH mode changes are in a downward direction (to a safer mode) and should not be preempted by an upward mode change request that happens to arrive on that cycle. Mode manager maintains a queue of any mode change requests that are generated during a given cycle, and then acts on the highest priority request at the end of the cycle. All lower-priority transition requests cause a status message to be generated, but the queue is flushed every cycle and they are never executed. All successfully executed mode changes generate status and event messages that go to the ground.

Figure 5: ACS Control Mode State Diagram



Autonomous Mode Changes

Autonomous mode changes are initiated from within mode manager and are distinguished from FDH-initiated transitions (which, strictly speaking, are also autonomous) because they can be upward to a higher mode. Also, they are built-in and cannot be easily disabled or reconfigured. There are two such transitions: the transition from TSA to STP, and the transition from MCAL back to the previous mode. The TSA-STP transition is conditional on acquiring the first timeline target, and the MCAL return transition occurs when the calibration is complete (approximately six seconds after entering MCAL). When either of these conditions is detected by the mode manager, a transition request to the appropriate mode of type "autonomous" is queued. If any FDH or ground-commanded request is queued on the same cycle, the autonomous transition request is rejected.

Ground-Commanded Mode Changes

Ground-commanded mode transition requests can be delivered to the ACS FSW no more than one per cycle. Only certain transitions are allowed by ground command, so these commands are first checked against an "allowed transition matrix" (see Table 5). For example, it is not allowed to jump directly from SCSSH to TSA or STP mode, as this would not make sense. If a commanded transition is not allowed, or if any FDH request is already on the queue, the command is rejected and a message to that effect is sent to the ground.

FDH-Initiated Mode Changes

If FDH detects a pointing constraint violation or other anomaly, it may trigger a safing transition to a lower control mode. Such action depends upon the particular FDH test being executed and the current control mode. FDH-initiated mode change requests are queued at the time the anomaly is detected, and they take precedence over both autonomous and ground-commanded mode change requests. There are four FDH tests that can cause mode changes (see Section V), so more than one such transition request may occur on a given cycle. In case of a conflict like this, the mode manager acts on the FDH mode change request that constitutes the most severe action (i.e., the one that initiates a change to the lowest mode). Status messages are sent to notify the ground of any FDH requests that are discarded in this way.

Setting Up for a New Mode

After resolving all conflicts and deciding what action to take, the mode manager sets the "current control mode" flag to its new value and executes the setup module for the current transition. This information is listed in Table 6. A possible generalization of this code would be to make each of these setup items an entry in a software table, so that the setup for a given transition could be reconfigurable from the ground. This design option was not exercised for WIRE due to its relatively few modes and the desire to fit the mode manager into pre-existing code.

Table 5: Mode Transitions Allowed by Ground Command

From/To	SCSSH	ZSP	TSA	STP	MCAL
SCSSH	Allowed	Allowed	Not	Not	Allowed
ZSP	Allowed	Allowed	Not*	Not	Allowed
TSA	Allowed	Allowed	Not	Not	Allowed
STP	Allowed	Allowed	Allowed	Not	Allowed
MCAL	Allowed	Allowed	Not	Not	Allowed

* Special case where transition is caused by two ground commands (Timeline Enable Command and valid Science Timeline Command) -- not a Mode Transition Command

Table 6: New Mode Setup Modules

From/To	SCS SH	ZSP	TSA	STP	MCAL
SCS SH	disable timeline, star acq proc off, set target type to autotarget	init struct filter state, set target type to autotarget, init coarse att solution, reset timeline	N/A	N/A	set MCAL step counter to 1, freeze RW rates, save prev mode, star acq proc off, reset MCAL complete flag
ZSP	disable timeline, star acq proc off, set target type to autotarget	init struct filter state, set target type to autotarget, init coarse att solution, reset timeline	init coarse att solution, set star acq RFOV values to coarse, reset timeline	N/A	set MCAL step counter to 1, freeze RW rates, save prev mode, star acq proc off, reset MCAL complete flag
TSA	disable timeline, star acq proc off, set target type to autotarget	init struct filter state, set target type to autotarget, init coarse att solution, reset timeline, star acq proc off	N/A	init fine attd solution	set MCAL step counter to 1, freeze RW rates, save prev mode, star acq proc off, reset MCAL complete flag
STP	disable timeline, star acq proc off, target type set to autotarget	init struct filter state, set target type to autotarget, init coarse att solution, reset timeline, star acq proc off	init coarse att solution, set star acq RFOV values to coarse, reset timeline, init struct filter state	N/A	set MCAL step counter to 1, freeze RW rates, save prev mode, star acq proc off, reset MCAL complete flag
MCAL	disable timeline, star acq proc off, target type set to autotarget, notify ground if left MCAL early	init struct filter state, set target type to autotarget, init coarse att solution, reset timeline, notify ground if left MCAL early	init coarse att solution, set star acq RFOV values to coarse, reset timeline, notify ground if left MCAL early	init fine attd solution, notify ground if left MCAL early	set MCAL step counter to 1, freeze RW rates, star acq proc off, reset MCAL complete flag

For all transitions: save previous mode, notify ground of transition and any overridden transitions

V – ACS Failure Detection and Handling Design

FDH Module Overview

For the most part, the WIRE ACS is a single-string design with a minimum number of redundant components, but the FSW does contain limited failure checking to monitor the health and performance of critical system components. The Failure Detection and Handling (FDH) module is comprised of several autonomous safety checks that enforce WIRE pointing constraints, handle hardware failures, and insure that certain processes perform within specified limits. If a violation condition is detected, FDH can cause mode changes, switch to a secondary hardware complement, increment statistic counters, and post event messages to aid in trouble shooting. These actions are designed to speed the return to the primary science mission or at least to place the spacecraft in a safe and stable configuration while the ground operators diagnose a failure condition. Autonomous failure recovery is important due to WIRE's infrequent ground contact schedule and short mission life.

The FDH logic is actually broken into two modules; the first monitors the health of sensors and actuators while the second checks to be sure the attitude control algorithms are behaving as expected. The first module, containing Y-RW, WAES, and gyro tests, runs immediately following the sensor data processing module. The second module, containing avoidance constraint, star acquisition, and attitude determination checks, runs immediately after the attitude control module (see Figure 4). This way, checks are performed using the most recent input data, and failures are detected as soon as they occur. Whether each check executes on a given cycle often depends on the current control mode and/or other high-level state flags.

It is important to note that if the ACE is in control of the spacecraft, the ACS FSW is still executing, but no FDH is run. ACESH mode is the spacecraft's insurance against a spurious FDH action based on bad information. Also, each FDH check may be individually disabled by the ground. For contingency situations, these safeguards provide a way for ground operators to prevent an autonomous mode change or reconfiguration to a state that is known to be undesirable.

Y-Reaction Wheel Failure Check

This FDH check monitors the health of the Y-reaction wheel. It is performed in SCSSH mode when the Y-wheel is responsible for providing Y-axis control. The check is made indirectly by ensuring that the control attitude about the Y-axis stays within a specified tolerance. A failure to control the Y-axis of the spacecraft will ultimately be detected by noting an unexpectedly large zenith error signal. It is possible that some fault other than a Y-wheel failure may be the cause of such a control failure, but resolution of this ambiguity is not necessary, since the only corrective action taken is to reconfigure SCSSH mode for three-wheel (ABC-wheel) control. This action is a one-way switch; once the reconfiguration is done, the ground must intervene to restore the original configuration and reset Y-wheel FDH. This test is not executed if it has already failed, if the current mode is not SCSSH, or if there is not currently a valid zenith error signal (supplied by either the WAES or the TriPAD routine). This test may also be disabled by ground command.

WAES Failure Check

This FDH check monitors the health of the wide-angle Earth sensor. Its output, the zenith error signal, is used for Earth avoidance control in SCSSH mode and for Earth avoidance constraint checking in FDH. The WAES check is done by comparing the zenith error signal from the WAES with that produced by the TriPAD routine. If these angles differ by more than a specified tolerance, the WAES is marked as failed and the software switches to using the TriPAD method of zenith angle determination. This test is not executed if it has already failed, if the spacecraft has entered eclipse, if an invalid magnetic field is sensed, if the sensed Sun and magnetic field vectors are too closely coaligned, or if the magnetic field model is not currently valid (conditions which preclude running the TriPAD routine). This test may also be disabled by ground command.

Gyro Acceleration Limit Check

This FDH check ensures the reasonableness of gyro readings. The difference between consecutive gyro rate readings is compared against the product of the expected angular acceleration and the elapsed time

between the two readings. If they don't compare within an allotted tolerance, the new reading is discarded and the previous reading is used on that control cycle. This test is executed only in ZSP, TSA, and STP modes (where the data is actually used). It may be disabled by ground command, but unlike most FDH tests, this one does not disable itself after taking action.

Sun Angle Violation Check

WIRE's short mission life and limited cryogen supply motivate a requirement to keep the boresight of the telescope at least 75 degrees from the Sun during all modes. This avoidance constraint is designed to prevent catastrophic instrument damage. Due to differences in control response, constraint checking is slightly different for each control mode, but a detected violation always results in a downward mode transition. For all FDH-initiated mode transitions, the goal is to fall down to the highest possible safe mode. This avoids time-costly overreactions while still keeping the telescope pointing in a safe direction.

Each control cycle, the x- and z-components of the Sun vector (as measured by the DSS or CSS in the spacecraft's body frame) are checked to see if they are outside the angle constraints designated for the current mode. If there is a violation, a counter is incremented; if no error condition exists, the violation counter is reset to zero. Then the counter is checked against the maximum count tolerance for the current mode. If the limit is exceeded, a constraint violation is declared.

When a constraint is violated, FDH posts an event message to the ground and submits a safing mode change request to the mode manager. If the current mode is STP or TSA, the requested mode is ZSP. If the current mode is ZSP, the requested mode is SCSSH. If the current mode is SCSSH, the ACS FSW gives up control of the spacecraft to the ACE box, and ACESH is entered. In the event of a persistent violation, each mode gets a chance to resolve the problem. Tolerances and timeouts for each mode are chosen with an eye towards giving each downward transition a reasonable time to recover safe pointing of the instrument without actually violating the hard constraint for any significant time. Angle and counter limits are

maintained in a table so they may be easily modified by ground operators.

Earth Angle Violation Check

WIRE also has a science-imposed Earth avoidance constraint, driven by the need to maximize the lifetime of the cryogen supply. This constraint minimizes the heat load on the cryostat by keeping the boresight of the telescope within 30 degrees of the zenith vector. Very much like the Sun avoidance check, the Earth avoidance check is control mode dependent, and a detected violation always results in a downward mode transition.

For the Earth constraint check, not only the limits and timeouts vary by mode; the method of computing the violation criterion varies, too. The WAES supplies a zenith error signal which is used to execute this constraint check in SCSSH and ZSP modes. This signal is the angle difference between the spacecraft boresight vector and the projection of the zenith vector onto the spacecraft x-z plane. In the event of a bad WAES signal, a backup zenith error signal is provided by the TriPAD method (an ephemeris and sensor based computation of the zenith error signal). If there is no valid ephemeris loaded, the zenith error signal is set to zero. In TSA and STP modes, an ephemeris is always available, so a more accurate constraint check can be made. In these modes, a cone angle is computed using the zenith error signal, spacecraft position vector, and Sun vectors from the models and DSS.

Each control cycle, the zenith error signal (or cone angle) is checked to see if it is outside the angle constraints designated for the current mode. If there is a violation, a counter is incremented; if no error condition exists, the violation counter is reset to zero. Then the counter is checked against the maximum count tolerance for the current mode. If the limit is exceeded, a constraint violation is declared. Violation handling is identical to that of the Sun angle check. A message is posted and a downward mode change is requested. Angle and counter limits are maintained in a table so they may be easily modified by ground operators.

Star Acquisition Timeout Check

This FDH check ensures that the star acquisition process has not taken longer than the time allotted for acquiring the current target. If such a failure is detected in STP mode, the star search is aborted and restarted in TSA mode with a larger search window. This gives the software a second chance to acquire a target in the event that the first attempt failed due to larger than expected fine attitude inaccuracies. If a failure is detected in TSA mode, a full field-of-view (FOV) search is performed and the current target is aborted. In this event, the current attitude is maintained and the timeline resumes with the next target. Ground operators can use data from the full FOV search to help determine why the target did not acquire.

Good Star Condition Check

This FDH check ensures that the “good star” criterion has not been persistently violated. Good star condition is a measure of the separation of the guide stars being used to track the current target, and thus, a measure of how well the stars can be used to update the fine attitude solution. Since good star condition is required to enter STP mode, this test is only executed in STP mode. A violation may occur if previously tracked stars are lost (moved off the edge of the tracker FOV or dropped by the tracker itself – a loss-of-track condition, or “LOT”). The fewer the number of tracked stars, the more difficult it is to satisfy the criterion. Each violation increments a statistic, but persistent violations will cause a transition back to TSA mode only if all the guide stars have been dropped.

Attitude Filter Covariance Checks

This FDH component consists of two separate checks which ensure that the coarse and fine attitude solutions have not diverged. Each covariance matrix is checked, and in each case a statistic is incremented if the matrix is not positive definite.

Attitude Filter Divergence Checks

This FDH component consists of two separate checks which ensure that the various on-board attitude

solutions have not diverged from each other. The coarse and TRIAD attitude solutions are compared in ZSP, TSA, and STP modes (and SCSSH if the coarse attitude override has been enabled). If these quaternions differ by more than a specified tolerance, a statistic is incremented; a persistent divergence will cause a mode transition to SCSSH mode (a bad gyro is assumed). In STP mode, the fine and TRIAD solutions are compared, and a statistic is incremented for each cycle on which the quaternions are judged to be too far apart.

VI – ACS Star Acquisition

The star acquisition process is based on the SWAS algorithm, but has been modified to perform under WIRE's more difficult acquisition requirements³. For SWAS, special predefined targets were used for entering the timeline, and these targets were chosen to be easily acquired using the coarse attitude solution. WIRE's Sun and Earth constraints preclude this approach and require the capability to enter the timeline using any science target.

For both SWAS and WIRE, the basic approach is to identify one star in a reduced field of view (RFOV) window as the base star, and then use that information to find the rest of the stars on the guide star list. But SWAS's direct match approach for base star identification has been replaced by a two-star approach. For WIRE, the base star is not verified until, using its observed location, a second guide star is found with the expected magnitude and relative location. These two stars are then used together in finding the remaining guide stars.

Logic was also added to maintain an array of up to four base star candidates (stars found in the RFOV with magnitudes similar to the base star). These candidates are then scrutinized in turn to find the one which has a matching second star from the guide star list. Any star in the guide star list can be used as the base star; the software starts with the first guide star, and later cycles through the others if it fails to be verified.

Another change made was to add logic to handle the case where a previously tracked star is suddenly

dropped by the CT-601. This situation has been reported on the Rossi X-Ray Timing Explorer (RXTE) mission, which uses the same star tracker. On detecting such dropouts, the WIRE FSW will resend the appropriate directed search command(s) in an attempt to reacquire the lost star(s).

The modifications made to SWAS's guide star acquisition algorithm are an attempt to minimize the risk of missing an initial target due to the arrangement of stars in the FOV, and to minimize the time spent reentering the timeline in the event that it is interrupted for any reason.

VII – ACS Tables

Table-driven Design Overview

A notable change in the ACS FSW from SWAS/TRACE to WIRE was the addition of 22 ground-modifiable tables containing parameters which had previously been hard-coded in the software or configurable only by individual ground command. These tables group related parameters such as control gains, sensor and actuator scale factors and biases, FDH limits and tolerances, and system enable flags. Tables may be modified temporarily by loading new values to RAM, or permanently by loading to EEPROM. The design philosophy was to place in tables those control parameters which are infrequently modified; system flags that are changed as part of regular operational procedures were left as command-modified values.

Software Design Changes to Accommodate Tables

Implementing the new table scheme required some important changes to the ACS software. Logic was added to provide an interface with the C&DH subsystem's Software Manager (SM) task, which provides utility routines for loading, dumping, and committing tables. The SM functionality was already in place for SWAS and TRACE, and in fact the C&DH subsystem itself is table-driven, but the ACS end of the interface had to be added.

At startup (after power up or cold restart), the ACS FSW uses a function provided by SM to initialize its

tables by copying them out of EEPROM to local RAM. Ground-initiated table loads to RAM are routed through SM, but executed by AC. AC is notified by SM that a new table load has arrived; then it calls an SM function to copy the data from a load buffer into its own RAM space. AC then sends a notification to SM that the transfer has completed. This mechanism allows the ACS software to retain control of when data is written into its data space. Responsibility for ACS table integrity (checksumming) as well as dumping table contents to the ground remains with C&DH since these operations do not involve modification of the active ACS data areas. Loads to the EEPROM versions of ACS tables are also handled exclusively by C&DH.

The switch to a table-driven scheme also required some reorganization of existing data structures, regrouping parameters from previously separate structures into tables organized by sensor/actuator and control mode. In addition, dynamic variables were segregated from constant variables; the ground-modifiable tables are entirely static (i.e. unchanged by the FSW) with one exception: the magnetic calibration table, which may be updated after MCAL mode computes its calibration of the TAM contamination matrix.

Finally, most of the pre-existing commands for changing individual FSW parameters and for enabling/disabling certain system functions were deleted. The number of ACS ground commands was cut from 123 for TRACE to 40 for WIRE.

Advantages

The use of tables in the WIRE ACS FSW allows many more software parameters to be modified, temporarily or permanently, and for the most part has simplified operational procedures by reducing the number of commands (see Table 7). Multiple related changes can be made with one table load rather than through a series of commands. Values which previously could be temporarily changed only by writing to RAM or permanently changed only by loading a complete software patch (both of which require detailed knowledge of the software map) can now be changed by a relatively simple table load to RAM or EEPROM. This capability is especially useful for ACS parameters which remain

undetermined until shortly before launch, such as sensor alignments and calibration measurements.

In addition, the state of the software configuration after a series of changes can be more easily verified in an almost automated fashion by dumping and comparing tables to original default images rather than reviewing hundreds of separate telemetry points.

Table 7: Table Design Considerations

Advantages	Disadvantages
Fewer commands to implement & test in FSW	Must carefully manage table loads, (on-orbit & testing)
Fewer commands in database	New ground tools & operations training required
Fewer commands necessary to configure software	Essentially requires second database to track table defaults
More configurable parameters (RAM/ROM loads)	Must modify code that might otherwise remain unchanged
RAM & EEPROM changes don't require code patch	Must modify inherited test procedures

Disadvantages

In general, one operational disadvantage of a table-driven architecture is that a change to any single element in a table requires uploading the entire table (partial table loads are possible, but cumbersome and unnecessarily risky, at least given WIRE's toolset). Unintended changes to other elements in a table may occur, especially if the particular table was modified by a previous load (prior changes may be accidentally undone). To mitigate this risk, table sizes are kept reasonable, related elements are grouped, and table loads are carefully managed; the defaults for each table and any changes made are tracked in a special database. For WIRE, table loads are expected to occur infrequently during actual spacecraft operations, but this issue can complicate spacecraft testing. For test procedures that load multiple tables (or versions of the same table) to set up desired test scenarios, special care must be taken to manage the loads. Upon completion of each test, care must also be taken to restore the default state of each modified table.

The conversion to the table-driven software required a new set of ground system tools for building and maintaining tables. This overhead also applies to the operations arena, where mission training is an issue. A WIRE-specific drawback to using tables was that a significant effort had to be expended in updating the set of test procedures inherited from earlier SMEX missions. A minor effort was also made to modify data structures in inherited software.

VIII - Conclusion

The WIRE ACS FSW evolved from previous SMEX implementations to meet the requirements imposed by its science objectives. It contains modules for data processing, attitude determination and control, guide star acquisition, actuator command generation, command and telemetry processing, and FDH. New features for WIRE include the ACS mode manager, which organizes and controls all other modules and dictates which software components are currently active. Significant additions and modifications were made in the FDH and star acquisition modules, and in making the software table-driven. The ACS FSW satisfies all target acquisition and pointing accuracy requirements, enforces Sun and Earth pointing constraints, provides the ground with a simple means for reconfiguration via table load, and meets all the demands of its real-time embedded environment.

IX - Acknowledgments

The authors wish to thank the following managers, subsystem engineers, algorithm developers, analysts, and software testers for their valuable technical input during the design and implementation of the WIRE ACS Flight Software: Cheryl L. Albert, Mark O. Anderson, Mike D. Blau, Jim Blue, Thomas J. Budney, J. Todd Campa, J. Roger Chen, Joel A. Chiralo, Thomas E. Correll, Jeffrey M. D'Agostino, Mary Jo Duncan, David F. Everett, Michael D. Fennell, Greg Greer, Taylor M. Hale, Tawanda M. Jacobs, Teresa L. Lafourcade, Kenneth L. Lebsock, Michael H. Lee, Ken Loo, Loc X. Luu, John A. McElvaney, Kent F. Mitterer, Robert Rapp, William M. Reid, Mark P. Richardson, Maxine R. Russell, Larry V. Shackelford, Timothy E. Singletary, Eric T. Stoneking, Christopher J. Thorpe, Victoriano Z. Untalan, and Andrew W. Wnuk.

X - Acronym List

AC	Attitude Control (software task)
ACE	Attitude Control Electronics
ACESH	Attitude Control Electronics Safehold mode
ACS	Attitude Control System
AL	Attitude Control Math Library
AM	Attitude Models (software task)
BC	1553 Bus Controller
C&DH	Command and Data Handling
CCD	Charged Coupled Device
CSS	Coarse Sun Sensor
CI	Command Ingest (software task)
CT-601	Star Tracker
DS	Directed Search, Data Storage (software task)
DSS	Digital Sun Sensor
ECI	Earth Centered Inertial
EDAC	Error Detection and Correction
FDH	Failure Detection and Handling
FOV	Field of View
FSW	Flight Software
GSFC	Goddard Space Flight Center
HK	Housekeeping (software task)
IC	Instrument Controller (software task)
I/O	Input/Output
LOT	Loss of Track
MCAL	Magnetic Calibration mode
MTB	Magnetic Torquer Bar
NASA	National Aeronautics and Space Administration
PROM	Programmable Read-Only Memory
RAM	Random Access Memory
RFOV	Reduced Field of View
RW	Reaction Wheel
SB	Software Bus (software task)
SC	Stored Command (software task)
SCS	Spacecraft Computer System
SCSSH	Spacecraft Computer System Safehold mode
SH	Scheduler (software task)
SM	Software Manager (software task)
SMEX	Small Explorer
SPE	Spacecraft Power Electronics
STP	Stellar Point mode
SWAS	Submillimeter Wave Astronomy Satellite
TAM	Three-Axis Magnetometer
TO	Telemetry Output (software task)
TRACE	Transition Region and Coronal Explorer
TriPAD	TRIAD Pitch Attitude Determination
TSA	Transitional Stellar Acquisition mode
WAES	Wide-Angle Earth Sensor
WIE	WIRE Instrument Electronics
WIRE	Wide-Field Infrared Explorer
RXTE	Rossi X-Ray Timing Explorer
ZSP	Zenith Sunpoint mode

XI - References

1. M. Fennell, V. Untalan, and M. Lee, "The Attitude Control System Design for the Wide-Field Infrared Explorer," Proceedings of the 11th Annual AIAA/USU Conference on Small Satellites, September 1997.
2. M. Anderson, M. Wennersten, J. Bonnet, and T.A. Hill, "The Submillimeter Wave Astronomy Satellite Attitude Control Software Design," Proceedings of the 9th Annual AIAA/USU Conference on Small Satellites, September 1995.
3. E. Stoneking, "A Robust Star Acquisition Algorithm for the Wide-Field Infrared Explorer," Proceedings of the 13th Annual AIAA/GSFC Conference on Space Flight Dynamics, May 1998.

XII – Author Biographies

Kenneth Barnes graduated from the University of Maine (BSEE) in 1990 and The George Washington University (MSEE) in 1992. He has worked at NASA's Goddard Space Flight Center for eight years, six of them as a software engineer for the Hammers Company, Inc. Since December 1995, Mr. Barnes has been working on the Attitude Control System (ACS) flight software for the Wide-Field Infrared Explorer (WIRE). Prior to that, he spent three years working on the ACS flight software for the Tropical Rainfall Measurement Mission (TRMM). While attending GWU, he was a PREST (Program for Research and Education in the Space Technologies) fellowship recipient, and worked part-time in NASA's Intelligent Robotics Laboratory on a global path-planning research project.

Charles Melhorn graduated from the University of Maryland, Baltimore County, in 1987 with a BA in Economics and in 1994 with a BS in Mathematics. He has been involved with flight software testing and development for multiple Small Explorer Project (SMEX) satellites at NASA's Goddard Space Flight Center since 1992. Mr. Melhorn worked on Command and Data Handling software build testing

for the Solar Anomalous Magnetospheric Particle Explorer (SAMPEX), flight software build and acceptance testing for the Fast Auroral Snapshot Explorer (FAST), Attitude Control System (ACS) flight software development and testing for the Transition Region and Coronal Explorer (TRACE), and ACS flight software development and testing for the Wide-Field Infrared Explorer (WIRE).

Tom Phillips graduated from Davis & Elkins College with a BSCS in 1986. He has been writing software professionally for 12 years. His aerospace software experiences include developing ground support software and attitude control software. His ACS software development experience includes NASA's Wide-Field Infrared Explorer (WIRE), Tropical Rainfall Measurement Mission (TRMM), Rossi X-Ray Timing Explorer (RXTE), Small Explorer – Lite (SMEX-LITE), and Spartan program. He also worked on hardware verification software for the Air Force STEP program. Prior to coming to NASA's Goddard Space Flight Center in 1992, Mr. Phillips worked with the U.S. Navy, developing shore-based communication software.